Final Project: Is AIS an Effective Tool for Ships to Avoid Whale Watch Boat Aggregtions?

Laura Howes

2018 Decemeber

# INTRODUCTION

#### Ship strike is one of the leading causes of serious injury and mortality to large whales (Thomas et al. 2015). While there have been many measures to mitigate this problem, ship strikes continue (van Der Hoop et al. 2013). For example, the critically endangered North Atlantic right whales are unresponsive to ship noise and have either negative or no reactions to alerting stimuli (Nowacek et al. 2004). Nowacek et al. (2004) found that alerting stimuli from vessels cause right whales to come up to the surface more frequently, proving greater risk to getting hit by a vessel. Large ship operators have utilized marine mammal observers, but observers are limited by personnel and type of ship.

#### Automatic Identification System (AIS) is a technology that provides ship position reports with location, time, and identification information. All ocean-going commercial traffic greater than 300 tons, vessels carrying 165 or more passengers, and tugs are required to carry AIS transmitters (Federal Register 2003; IALA 2004). NOAA’s Stellwagen Bank National Marine Sanctuary (SBNMS) installed 3 AIS receivers near Massachusetts Bay to monitor ship traffic transiting the SBNMS. Using this data, record of ship traffic patterns can be described, including commercial whale watch boats. AIS data have been a useful tool in the SBNMS for conservation management, effectively reducing the risk of ship strike (Wiley et al. 2013). However, as ship strike persists and exists in other areas throughout the world, and as new shifts in large whale distributions are observed (Davis et al. 2017), additional measures to protect large whales are still needed.

#### The question of this study is to investigate whether large commercial ships change their traffic patterns when they detect whale watch boats via AIS in the Boston Traffic Separation Scheme (TSS). If ships deviated from their intended track due to whale watch boats detected on AIS, this essentially helps create a buffer for large whale aggregations during the whale watch season.

# METHODS

### Model Creation

#### To measure whether ships are changing their track line, deviations from their outgoing/ingoing travel path needed to be measured. Predictors that would show ship operator reactions would be: 1) Presence of AIS on whale watch boats, 2) Time of day, 3) Turning of the ship, 4) Change in velocity, and 5) Distance from whale watch boats when ship starts to turn/change velocity

#### Visualization of model and predictors is show in Figure 1:

setwd(“C:/Users/Laura/Dropbox/Grad school/BIOL 607 Biostats/Final Project/Figures”) getwd()

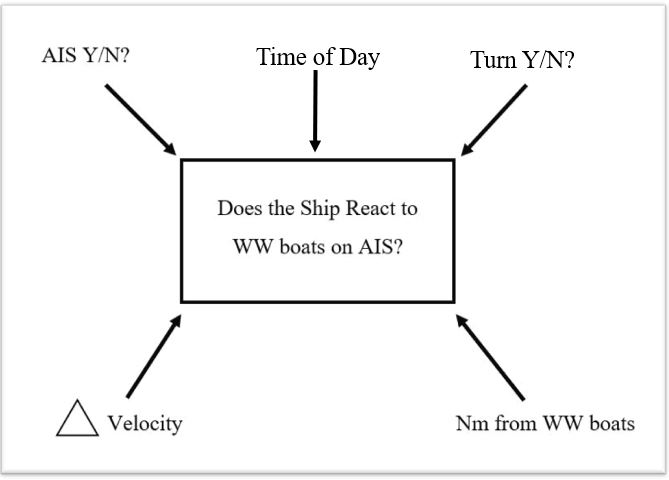


Figure 1 - Model Design

### Simulated Data

#### As this was an experimental design, simulated data was produced via RStudio as seen in Figure 2. The control group, named “WW\_AIS\_N”, consisted of days without whale watch boats reported via AIS in the TSS. The experimental group, “WW\_AIS\_Y”, consisted of days with whale watch boats reported via AIS. Simulated data assumed a normal distribution and reflected predicted trends in the TSS. “Time of day” column, labeled “Time” assumed a random distribution and was divided into 4 groups: 8:00-10:59 (morning), 2 = 11:00-13:59 (midday), 3 = 14:00- 16:59 (afternoon), 17:00 - 19:59 (evening), as time is not predicted to have a significant effect since whale watch boats only operate during daylight hours. Change in velocity data is referred to as “Knot” in the table and this study, “Turn Y/N” was referred to as “Angle”, and “Distance from whale watch boats was referred to as”Nm\_from\_boat“.

set.seed(2018)  
  
ship\_data\_rnorm\_Y <- data.frame(WW\_AIS = sample(c("Y"), 100, replace = TRUE),angle = (rnorm(100, 45, 10)),knots = (rnorm(100, 10, 3)),time = sample(c(1:4), 100, replace = TRUE),nm\_from\_boat = (rnorm(100, 15, 5)))  
  
ship\_data\_rnorm\_N <- data.frame(WW\_AIS = sample(c("N"), 100, replace = TRUE),angle = (rnorm(100, 0, .5)),knots = (rnorm(100, 5, 1)),time = sample(c(1:4), 100, replace = TRUE),nm\_from\_boat = (rnorm(100, 0, 0)))  
  
#combined data  
ship\_data <- rbind(ship\_data\_rnorm\_Y, ship\_data\_rnorm\_N)  
  
ship\_data

## WW\_AIS angle knots time nm\_from\_boat  
## 1 Y 39.125572766 10.494202 4 7.668510  
## 2 Y 39.725563316 11.034203 4 14.360914  
## 3 Y 50.960211604 16.835251 4 17.503704  
## 4 Y 32.286875967 13.345833 1 14.879027  
## 5 Y 42.987454563 11.739091 4 13.681662  
## 6 Y 49.556339121 8.764072 1 20.652131  
## 7 Y 35.651730742 13.744754 1 15.215658  
## 8 Y 72.283183026 10.743079 4 15.059017  
## 9 Y 48.466448687 12.010125 4 15.760441  
## 10 Y 38.622902887 5.924038 2 11.365462  
## 11 Y 53.351829749 7.521381 4 20.245179  
## 12 Y 50.890225689 12.480510 2 19.484454  
## 13 Y 46.700937735 15.329943 3 16.423575  
## 14 Y 40.628284105 14.155676 2 13.652162  
## 15 Y 44.197919394 5.298794 3 10.915027  
## 16 Y 35.724572386 6.326753 1 11.703812  
## 17 Y 60.188671470 13.045032 1 15.911864  
## 18 Y 45.857304515 12.402579 2 12.382032  
## 19 Y 66.527947553 6.572899 3 21.847482  
## 20 Y 31.340685048 11.709953 2 13.005878  
## 21 Y 48.459062255 14.653883 1 19.604247  
## 22 Y 50.651806995 11.052958 4 3.031156  
## 23 Y 44.939021630 8.355113 3 18.627762  
## 24 Y 40.612189344 13.537609 1 19.736085  
## 25 Y 51.888910965 7.935605 4 11.999662  
## 26 Y 40.005808050 5.052178 1 14.620448  
## 27 Y 26.313853705 7.197050 4 20.933887  
## 28 Y 52.434253889 11.394741 2 15.881385  
## 29 Y 44.108250715 7.117681 1 14.376380  
## 30 Y 55.811636829 8.481446 3 5.388341  
## 31 Y 31.353952952 13.598503 1 11.689551  
## 32 Y 61.490110545 8.704035 4 15.816596  
## 33 Y 37.207436053 9.726277 3 25.031456  
## 34 Y 54.831038168 8.661395 3 14.405111  
## 35 Y 50.057835523 17.106911 4 9.721235  
## 36 Y 28.002850087 10.409553 4 21.048550  
## 37 Y 70.974628221 7.039186 1 22.787273  
## 38 Y 40.773439247 12.449791 2 11.287325  
## 39 Y 37.664629670 5.274658 1 20.387974  
## 40 Y 47.759453826 13.187282 3 19.195771  
## 41 Y 35.174777416 9.879913 4 9.757005  
## 42 Y 51.547661648 7.866120 2 15.160956  
## 43 Y 59.740871778 10.800102 1 6.630802  
## 44 Y 47.129818970 13.142107 3 17.369811  
## 45 Y 29.506299389 11.175307 1 15.148891  
## 46 Y 47.771736266 15.250965 3 20.972054  
## 47 Y 54.832101490 10.476564 3 18.251584  
## 48 Y 52.593880091 10.398284 4 14.697972  
## 49 Y 38.426452290 6.656386 3 8.418802  
## 50 Y 54.229037439 12.697751 2 6.573152  
## 51 Y 48.709081873 11.849004 1 15.656389  
## 52 Y 51.733371342 5.130806 4 15.555076  
## 53 Y 37.774470234 13.810591 3 17.261166  
## 54 Y 39.291252730 13.149986 2 25.834165  
## 55 Y 42.303854132 11.352264 4 15.340690  
## 56 Y 53.043303772 15.690722 1 11.369822  
## 57 Y 32.788912040 6.318431 4 12.941217  
## 58 Y 55.742602350 11.735559 1 20.916937  
## 59 Y 51.518996803 10.634669 1 17.184328  
## 60 Y 46.697824034 9.320746 1 12.787613  
## 61 Y 39.364448676 8.145905 1 10.029912  
## 62 Y 24.042530588 5.998367 4 13.670765  
## 63 Y 49.900030337 12.059842 1 7.009943  
## 64 Y 50.789596483 10.544502 1 14.188945  
## 65 Y 40.606540921 10.890074 3 16.188693  
## 66 Y 47.217339290 15.055339 2 8.049747  
## 67 Y 27.334063247 13.613587 4 22.025102  
## 68 Y 35.735328793 11.403317 1 12.783464  
## 69 Y 74.559451550 10.408119 2 11.169863  
## 70 Y 47.452776661 11.015589 3 14.623061  
## 71 Y 43.424713833 8.984554 2 11.615382  
## 72 Y 46.446171041 5.778767 3 14.745260  
## 73 Y 46.941759279 12.136536 4 16.204042  
## 74 Y 46.568753870 10.228603 2 15.542738  
## 75 Y 32.095896668 12.846895 4 14.605672  
## 76 Y 39.892435708 12.667288 2 10.495937  
## 77 Y 37.019358583 9.874871 1 20.074588  
## 78 Y 40.119730035 9.331556 3 12.771745  
## 79 Y 72.109987436 11.994714 3 16.046299  
## 80 Y 44.849492369 9.286704 2 7.526530  
## 81 Y 46.708882422 12.890320 2 14.430708  
## 82 Y 31.434601397 3.751445 3 11.399487  
## 83 Y 47.407596039 8.922495 2 12.550734  
## 84 Y 50.061862462 9.499021 2 15.629731  
## 85 Y 34.629018664 15.298461 1 18.030838  
## 86 Y 53.965343450 10.034783 3 13.356665  
## 87 Y 49.108859935 11.793512 1 22.383916  
## 88 Y 44.466833051 11.783492 1 19.471244  
## 89 Y 41.555688329 14.426354 1 27.558954  
## 90 Y 24.895922157 7.473003 4 21.053510  
## 91 Y 58.643279906 7.037573 1 19.847307  
## 92 Y 42.402988910 6.002718 4 16.066547  
## 93 Y 44.832232950 6.834746 3 6.647139  
## 94 Y 45.592768963 17.854420 1 10.195756  
## 95 Y 48.492083316 10.126969 3 11.211892  
## 96 Y 60.089901789 5.135144 3 18.436349  
## 97 Y 52.474377275 15.898731 2 15.843690  
## 98 Y 45.982363267 8.672387 4 7.235591  
## 99 Y 61.987040085 13.177116 3 14.061164  
## 100 Y 48.579820438 7.321432 4 14.553274  
## 101 N -1.125807801 5.163799 3 0.000000  
## 102 N -1.113958692 4.009219 1 0.000000  
## 103 N 0.153576815 3.617200 1 0.000000  
## 104 N 0.015048713 4.283838 2 0.000000  
## 105 N -0.728672098 4.385409 3 0.000000  
## 106 N -0.126290671 4.132734 3 0.000000  
## 107 N -0.167319231 5.528307 1 0.000000  
## 108 N 0.448524650 4.159643 4 0.000000  
## 109 N 0.034334350 5.269946 4 0.000000  
## 110 N 0.905067265 6.400324 1 0.000000  
## 111 N 0.824093869 5.694768 4 0.000000  
## 112 N 0.108744344 4.314617 1 0.000000  
## 113 N -0.793578634 4.798937 2 0.000000  
## 114 N 0.742571644 4.913274 4 0.000000  
## 115 N -0.845100406 3.863130 4 0.000000  
## 116 N 0.380909214 6.534119 3 0.000000  
## 117 N 0.094743908 6.165963 2 0.000000  
## 118 N 0.074800098 4.566904 1 0.000000  
## 119 N -0.076138949 4.302109 3 0.000000  
## 120 N 0.077255178 3.704801 1 0.000000  
## 121 N -0.175219593 6.494771 1 0.000000  
## 122 N -0.166274041 4.924239 1 0.000000  
## 123 N 0.028514885 3.693519 4 0.000000  
## 124 N -0.170236882 4.874206 1 0.000000  
## 125 N -1.167743064 5.893977 3 0.000000  
## 126 N 0.729365239 5.423989 4 0.000000  
## 127 N -0.554703838 4.956020 2 0.000000  
## 128 N -0.912452046 4.620379 4 0.000000  
## 129 N 0.296053647 3.707487 2 0.000000  
## 130 N -0.397076824 5.255459 3 0.000000  
## 131 N -0.589401167 7.071436 3 0.000000  
## 132 N -0.218023864 4.013178 4 0.000000  
## 133 N 0.020380024 7.634873 2 0.000000  
## 134 N -0.150290724 5.030044 2 0.000000  
## 135 N -1.060706691 4.029051 2 0.000000  
## 136 N 0.097403530 4.888993 3 0.000000  
## 137 N 0.739710638 4.874364 1 0.000000  
## 138 N -0.465987825 4.883840 1 0.000000  
## 139 N 0.235354750 4.858270 1 0.000000  
## 140 N 0.143546411 5.836770 3 0.000000  
## 141 N -0.279423981 5.266321 4 0.000000  
## 142 N -0.270885124 4.407783 4 0.000000  
## 143 N -0.179496117 5.713758 3 0.000000  
## 144 N -0.224437056 5.135692 4 0.000000  
## 145 N -0.365237638 3.569887 1 0.000000  
## 146 N 0.086706526 3.819100 4 0.000000  
## 147 N -0.002264478 5.759953 4 0.000000  
## 148 N -0.984589983 4.239234 4 0.000000  
## 149 N -0.105994804 5.315897 1 0.000000  
## 150 N -0.787991930 5.708887 1 0.000000  
## 151 N 0.528121140 7.325642 1 0.000000  
## 152 N -0.232313683 4.837182 1 0.000000  
## 153 N -0.086472110 4.834247 3 0.000000  
## 154 N -0.383801690 7.505741 1 0.000000  
## 155 N -0.011712091 5.419934 2 0.000000  
## 156 N 0.188442067 4.507654 2 0.000000  
## 157 N 0.202477252 5.947707 4 0.000000  
## 158 N 0.039442439 4.861037 4 0.000000  
## 159 N -0.584528492 5.537216 2 0.000000  
## 160 N -0.083477243 2.946054 4 0.000000  
## 161 N 0.335837766 5.608874 3 0.000000  
## 162 N 0.067232113 6.465787 3 0.000000  
## 163 N 0.223113705 6.465535 3 0.000000  
## 164 N -0.370975566 6.608990 3 0.000000  
## 165 N -0.430810611 5.074186 2 0.000000  
## 166 N 0.245719726 5.656035 4 0.000000  
## 167 N -0.416268230 5.125951 2 0.000000  
## 168 N -0.795315260 6.645149 4 0.000000  
## 169 N 0.159444873 4.123928 3 0.000000  
## 170 N -0.382046999 6.189844 4 0.000000  
## 171 N 0.537417403 5.512513 4 0.000000  
## 172 N 0.271769132 6.163590 4 0.000000  
## 173 N -0.410435485 5.266228 2 0.000000  
## 174 N 0.276661952 5.131117 3 0.000000  
## 175 N 0.389774043 5.969994 1 0.000000  
## 176 N 0.771681446 4.810194 4 0.000000  
## 177 N 0.066286287 3.709349 2 0.000000  
## 178 N -0.200321043 5.106583 1 0.000000  
## 179 N -0.813555778 6.392371 1 0.000000  
## 180 N -0.438325366 4.829586 2 0.000000  
## 181 N 0.730370243 2.996506 4 0.000000  
## 182 N -0.032518595 4.370909 2 0.000000  
## 183 N 0.787257549 4.875288 1 0.000000  
## 184 N 0.036405462 5.313938 3 0.000000  
## 185 N -0.769292215 5.433447 2 0.000000  
## 186 N 0.390962139 4.932594 2 0.000000  
## 187 N -0.654563690 4.133681 3 0.000000  
## 188 N 0.522305510 4.641571 2 0.000000  
## 189 N 0.201649892 5.611419 4 0.000000  
## 190 N -0.451183759 6.117251 3 0.000000  
## 191 N -0.428056448 4.836275 1 0.000000  
## 192 N 0.332827088 6.540938 1 0.000000  
## 193 N -0.671796626 3.911999 2 0.000000  
## 194 N -0.135536777 6.587681 1 0.000000  
## 195 N 0.673630199 5.245344 1 0.000000  
## 196 N 0.275693614 2.298236 2 0.000000  
## 197 N 0.284324359 2.854863 1 0.000000  
## 198 N -0.103927589 3.577607 4 0.000000  
## 199 N -0.080894266 5.189313 3 0.000000  
## 200 N -0.207840296 3.864804 3 0.000000

##### Figure 2 - Simulated Data set

### Statisical Testing

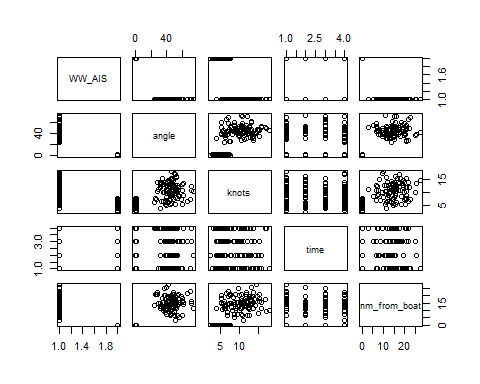
#### A Welch’s t-test was used to compare control and experimental means of each predictor. This variation of a two-sample t-test compares the means of two groups, is used when the variances of the two groups are unequal, and assumes normality of the data (Whitlock & Schluter 2015). An alpha of 0.05 was used to determine statistical significance, as it is the conventional standard, however p-values will not be as strongly supported in this study if they are in a 0.01-0.04 range under 0.05. As each predictor is assumed independent of each other via the simulated data set, multiple Welch’s t-tests were tested for each predictor. An optimal sample size for means and power analyses were performed for each predictor in order to calculate minimum and maximum sample sizes needed for significant results. For power, a beta was set at 0.08, as it is the conventional power threshold (Whitlock & Schluter 2015), although when performing analysis on real-time data, a threshold goal for beta will be 1.0.

# RESULTS

### Initial visualization

#### Figure 3 produces a multi-comparison of predictors to look for overall patterns. Box plots were used to compare control and experimental groups for each predictor show in Figure 4. At a visual glace, Angle, Knot, and Nm\_from\_boat predictors show different ranges and medians (except for Time, which was randomized).

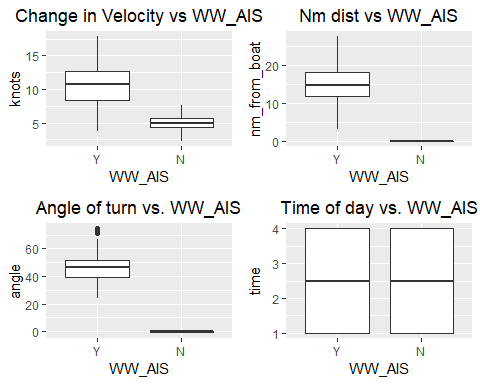
plot(ship\_data)



#multi-comparisons of predictors

##### Figure 3 - Initial visualization for spotting trends

knot <- ggplot(data = ship\_data, aes(x = WW\_AIS, y = knots)) +  
 geom\_boxplot() +  
 ggtitle("Change in Velocity vs WW\_AIS") +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
nm <- ggplot(data = ship\_data, aes(x = WW\_AIS, y = nm\_from\_boat)) +  
 geom\_boxplot() +  
 ggtitle("Nm dist vs WW\_AIS") +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
angle <- ggplot(data = ship\_data, aes(x = WW\_AIS, y = angle)) +  
 geom\_boxplot() +  
 ggtitle("Angle of turn vs. WW\_AIS") +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
time <- ggplot(data = ship\_data, aes(x = WW\_AIS, y = time)) +  
 geom\_boxplot() +  
 ggtitle("Time of day vs. WW\_AIS") +  
 theme(plot.title = element\_text(hjust = 0.5))  
  
knot + nm + angle + time



#Comparison of 4 predictors

##### Figure 4: Boxplot visualizations to compare medians and symmetry, of the 4 predictors Knot, Nm\_from\_boat, Angle, and Time.

### Means and Standard Deviations

#### Means and SD were calculated for later power analysis shown in Figures 5 & 6. To prepare data for Welch’s t-tests, data was spread by a factor of WW\_AIS Y and N (see Figure 5)

#separated data to test the two means  
  
ship\_data\_separated <- data.frame(angle\_Y = (rnorm(100, 45, 10)),knots\_Y = (rnorm(100, 10, 3)),time\_Y = sample(c(1:4), 100, replace = TRUE),nm\_from\_boat\_Y = (rnorm(100, 15, 5)),angle\_N = (rnorm(100, 0, .5)),knots\_N = (rnorm(100, 5, 1)),time\_N = sample(c(1:4), 100, replace = TRUE),nm\_from\_boat\_Y = (rnorm(100, 15, 5)),nm\_from\_boat\_N = (rnorm(100, 0, 0)))  
  
head(ship\_data\_separated)

## angle\_Y knots\_Y time\_Y nm\_from\_boat\_Y angle\_N knots\_N time\_N  
## 1 56.23350 7.098406 3 19.020250 0.25000625 6.246018 1  
## 2 59.33945 15.049130 1 23.517298 -0.20977316 4.596425 2  
## 3 42.01952 10.283225 2 16.074706 -0.01624121 4.437061 3  
## 4 34.54722 8.636858 1 6.040508 -0.84868282 4.417957 1  
## 5 40.66379 15.216230 1 15.562666 -0.30243193 4.232060 3  
## 6 49.31665 7.619218 1 19.081317 0.45092218 3.608371 3  
## nm\_from\_boat\_Y.1 nm\_from\_boat\_N  
## 1 10.31144 0  
## 2 21.83895 0  
## 3 17.19939 0  
## 4 10.66273 0  
## 5 15.36725 0  
## 6 12.86948 0

mean\_nm\_Y <- mean(ship\_data\_separated$nm\_from\_boat\_Y)  
mean\_angle\_Y <- mean(ship\_data\_separated$angle\_Y)  
mean\_knots\_Y <- mean(ship\_data\_separated$knots\_Y)  
mean\_time\_Y <- mean(ship\_data\_separated$time\_Y)  
  
mean\_nm\_N <- mean(ship\_data\_separated$nm\_from\_boat\_N)  
mean\_angle\_N <- mean(ship\_data\_separated$angle\_N)  
mean\_knots\_N <- mean(ship\_data\_separated$knots\_N)  
mean\_time\_N <- mean(ship\_data\_separated$time\_N)  
  
aggregate(nm\_from\_boat ~ WW\_AIS, ship\_data, mean)

## WW\_AIS nm\_from\_boat  
## 1 Y 14.94449  
## 2 N 0.00000

aggregate(knots ~ WW\_AIS, ship\_data, mean)

## WW\_AIS knots  
## 1 Y 10.539496  
## 2 N 5.056223

aggregate(angle ~ WW\_AIS, ship\_data, mean)

## WW\_AIS angle  
## 1 Y 45.90753005  
## 2 N -0.08605721

aggregate(time ~ WW\_AIS, ship\_data, mean)

## WW\_AIS time  
## 1 Y 2.46  
## 2 N 2.49

##### Figure 5: Calculation of means. Data frame was separated into Conrol and Experimental Groups by ww\_AIS “Y or N”.

sd\_nm\_Y <- sd(ship\_data\_separated$nm\_from\_boat\_Y)  
sd\_angle\_Y <-sd(ship\_data\_separated$angle\_Y)  
sd\_knots\_Y <- sd(ship\_data\_separated$knots\_Y)  
sd\_time\_Y <- sd(ship\_data\_separated$time\_Y)  
  
sd\_nm\_N <- sd(ship\_data\_separated$nm\_from\_boat\_N)  
sd\_angle\_N <- sd(ship\_data\_separated$angle\_N)  
sd\_knots\_N <-sd(ship\_data\_separated$knots\_N)  
sd\_time\_N <- sd(ship\_data\_separated$time\_N)  
  
aggregate(nm\_from\_boat ~ WW\_AIS, ship\_data, sd)

## WW\_AIS nm\_from\_boat  
## 1 Y 4.679525  
## 2 N 0.000000

aggregate(knots ~ WW\_AIS, ship\_data, sd)

## WW\_AIS knots  
## 1 Y 3.104718  
## 2 N 1.042770

aggregate(angle ~ WW\_AIS, ship\_data, sd)

## WW\_AIS angle  
## 1 Y 10.262830  
## 2 N 0.480541

aggregate(time ~ WW\_AIS, ship\_data, sd)

## WW\_AIS time  
## 1 Y 1.175679  
## 2 N 1.167705

##### Figure 6: Calculation of standard deviations

### Welch Two-sample t-test

#### All tests, except for the Time t-test, rejected the null that there was no significant difference between the control and experimental means. In Figure 7, p-value results for Angle, Knot, and Nm\_from\_boat were < 2.2 e-16, showing very high certainty. Time p-value = 0.4587, which accepts the null that there was no significant difference between the control and experimental means, was expected due to the structure of the simulated time data.

#Welch two-sample t-test  
t.test(ship\_data\_separated$nm\_from\_boat\_Y, ship\_data\_separated$nm\_from\_boat\_N)

##   
## Welch Two Sample t-test  
##   
## data: ship\_data\_separated$nm\_from\_boat\_Y and ship\_data\_separated$nm\_from\_boat\_N  
## t = 28.511, df = 99, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 13.91404 15.99556  
## sample estimates:  
## mean of x mean of y   
## 14.9548 0.0000

t.test(ship\_data\_separated$knots\_Y, ship\_data\_separated$knots\_N)

##   
## Welch Two Sample t-test  
##   
## data: ship\_data\_separated$knots\_Y and ship\_data\_separated$knots\_N  
## t = 17.163, df = 129.29, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 4.640591 5.849872  
## sample estimates:  
## mean of x mean of y   
## 10.191029 4.945798

t.test(ship\_data\_separated$angle\_Y, ship\_data\_separated$angle\_N)

##   
## Welch Two Sample t-test  
##   
## data: ship\_data\_separated$angle\_Y and ship\_data\_separated$angle\_N  
## t = 41.567, df = 99.47, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 42.63647 46.91082  
## sample estimates:  
## mean of x mean of y   
## 44.74778020 -0.02586434

t.test(ship\_data\_separated$time\_Y, ship\_data\_separated$time\_N)

##   
## Welch Two Sample t-test  
##   
## data: ship\_data\_separated$time\_Y and ship\_data\_separated$time\_N  
## t = -0.74252, df = 197.45, p-value = 0.4587  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.4021463 0.1821463  
## sample estimates:  
## mean of x mean of y   
## 2.42 2.53

tidy(t.test(ship\_data\_separated$nm\_from\_boat\_Y, ship\_data\_separated$nm\_from\_boat\_N))

## # A tibble: 1 x 10  
## estimate estimate1 estimate2 statistic p.value parameter conf.low  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 15.0 15.0 0 28.5 1.56e-49 99 13.9  
## # ... with 3 more variables: conf.high <dbl>, method <chr>,  
## # alternative <chr>

tidy(t.test(ship\_data\_separated$knots\_Y, ship\_data\_separated$knots\_N))

## # A tibble: 1 x 10  
## estimate estimate1 estimate2 statistic p.value parameter conf.low  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 5.25 10.2 4.95 17.2 3.87e-35 129. 4.64  
## # ... with 3 more variables: conf.high <dbl>, method <chr>,  
## # alternative <chr>

tidy(t.test(ship\_data\_separated$angle\_Y, ship\_data\_separated$angle\_N))

## # A tibble: 1 x 10  
## estimate estimate1 estimate2 statistic p.value parameter conf.low  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 44.8 44.7 -0.0259 41.6 1.11e-64 99.5 42.6  
## # ... with 3 more variables: conf.high <dbl>, method <chr>,  
## # alternative <chr>

tidy(t.test(ship\_data\_separated$time\_Y, ship\_data\_separated$time\_N))

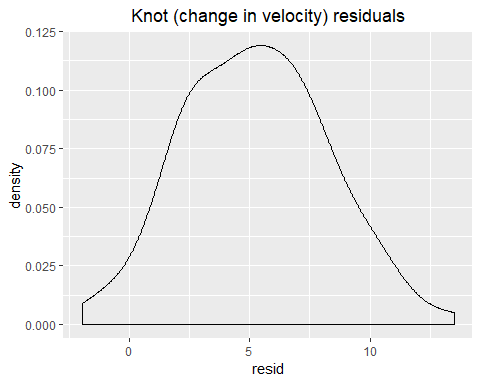
## # A tibble: 1 x 10  
## estimate estimate1 estimate2 statistic p.value parameter conf.low  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 -0.110 2.42 2.53 -0.743 0.459 197. -0.402  
## # ... with 3 more variables: conf.high <dbl>, method <chr>,  
## # alternative <chr>

##### Figure 7: Welch two-sample t-test results

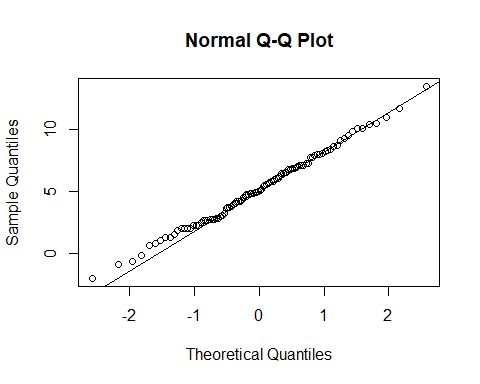
### Testing Assumption of Welch Two Sample t-test

#### To test normality of data, residuals were analyzed visually, plotted via Q-Q plots of sample vs. theoretical quantiles, and tested with the Shapiro-Wilk normality test. Figures 8-10 show normality for Angle, Knot, and Nm\_from\_boat predictors, as to be expected from simulated data structure. In the Shapiro-Wilk normality test, all three predictors respectively failed to reject the null hypothesis that the data are from a normal distribution, with p-values = 0.9804 (Knot), 0.06558 (Nm\_from\_boat), and 0.06581 (Angle). Nm\_from\_boat and Angle predictors hovered near an alpha of 0.05, hence the residual plots did not appear as smooth at Knot predictor. Figure 11 shows Time data was not normal (as to be expected), with the Shapiro-Wilk normality test p-value = 0.0006204. This rejects the null hypothesis that the data is normal.

###Knot data  
  
#Check residuals  
ship\_resid\_kt <- ship\_data\_separated %>%  
 mutate(resid = ship\_data\_separated$knots\_Y -  
 ship\_data\_separated$knots\_N)  
  
##evaluate residuals visually  
ggplot(ship\_resid\_kt,  
 mapping = aes(x= resid)) +  
 geom\_density() + #seems normal   
 ggtitle("Knot (change in velocity) residuals") +  
 theme(plot.title = element\_text(hjust = 0.5))



#qq plot  
qqnorm(ship\_resid\_kt$resid)  
qqline(ship\_resid\_kt$resid)



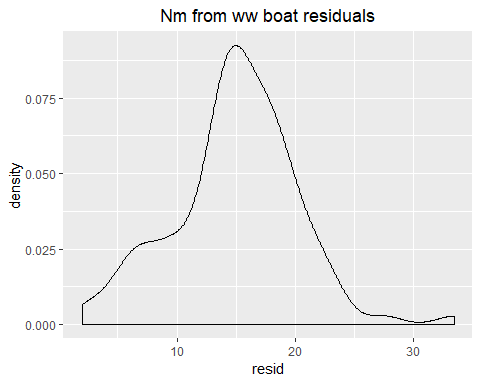
#put it to the test  
shapiro.test(ship\_resid\_kt$resid)

##   
## Shapiro-Wilk normality test  
##   
## data: ship\_resid\_kt$resid  
## W = 0.99522, p-value = 0.9804

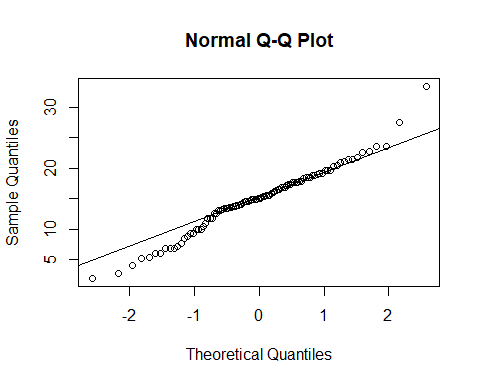
#W = 0.99522, p-value = 0.9804  
#failed to reject the null hypothesis, so it is normal

##### Figure 8: Testing assumptions of Welch two-sample t-test,Knot predictor data

###Nm\_from\_boat data  
  
#Check residuals  
ship\_resid\_nm <- ship\_data\_separated %>%  
 mutate(resid = ship\_data\_separated$nm\_from\_boat\_Y -  
 ship\_data\_separated$nm\_from\_boat\_N)   
  
##evaluate residuals visually  
ggplot(ship\_resid\_nm,  
 mapping = aes(x= resid)) +  
 geom\_density() + #not as great   
 ggtitle("Nm from ww boat residuals") +  
 theme(plot.title = element\_text(hjust = 0.5))



#qq plot  
qqnorm(ship\_resid\_nm$resid)  
qqline(ship\_resid\_nm$resid) #tails are off



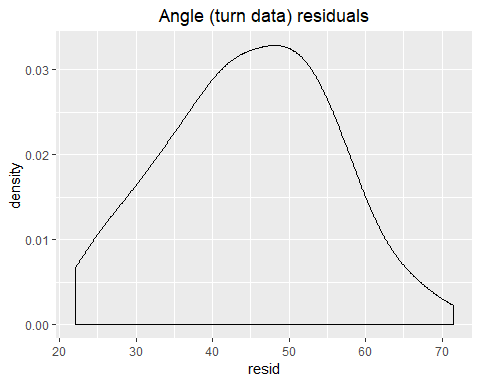
#put it to the test  
shapiro.test(ship\_resid\_nm$resid)

##   
## Shapiro-Wilk normality test  
##   
## data: ship\_resid\_nm$resid  
## W = 0.97608, p-value = 0.06558

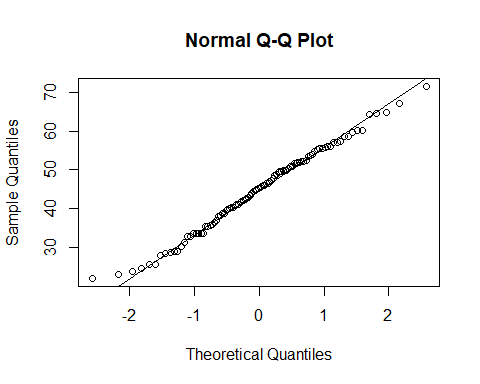
#W = 0.97608, p-value = 0.06558  
#failed to reject the null hypothesis, so it is normal

##### Figure 9: Testing assumptions of Welch two-sample t-test, Nm\_from\_boat data

###Angle data  
  
#Check residuals  
ship\_resid\_angle <- ship\_data\_separated %>%  
 mutate(resid = ship\_data\_separated$angle\_Y -  
 ship\_data\_separated$angle\_N)  
  
  
##evaluate residuals visually  
ggplot(ship\_resid\_angle,  
 mapping = aes(x= resid)) +  
 geom\_density() + #looks pretty good  
 ggtitle("Angle (turn data) residuals") +  
 theme(plot.title = element\_text(hjust = 0.5))



#qq plot  
qqnorm(ship\_resid\_angle$resid)  
qqline(ship\_resid\_angle$resid) #looks good



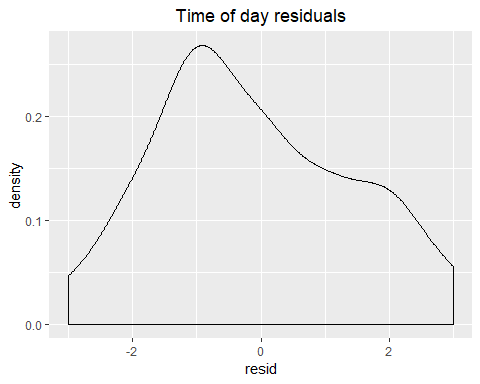
#put it to the test  
shapiro.test(ship\_resid\_angle$resid)

##   
## Shapiro-Wilk normality test  
##   
## data: ship\_resid\_angle$resid  
## W = 0.98992, p-value = 0.6581

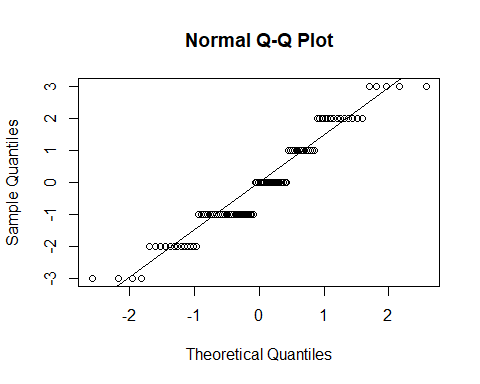
#W = 0.98992, p-value = 0.06581  
#failed to reject the null hypothesis, so it is normal

##### Figure 10: Testing assumptions of Welch two-sample t-test, Angle data

###Time data  
  
#Check residuals  
ship\_resid\_time <- ship\_data\_separated %>%  
 mutate(resid = ship\_data\_separated$time\_Y -  
 ship\_data\_separated$time\_N)  
  
  
##evaluate residuals visually  
ggplot(ship\_resid\_time,  
 mapping = aes(x= resid)) +  
 geom\_density() + #looks pretty good, weird hump  
 ggtitle("Time of day residuals") +  
 theme(plot.title = element\_text(hjust = 0.5))



#qq plot  
qqnorm(ship\_resid\_time$resid)  
qqline(ship\_resid\_time$resid) #funky



#put it to the test  
shapiro.test(ship\_resid\_time$resid)

##   
## Shapiro-Wilk normality test  
##   
## data: ship\_resid\_time$resid  
## W = 0.9374, p-value = 0.0001342

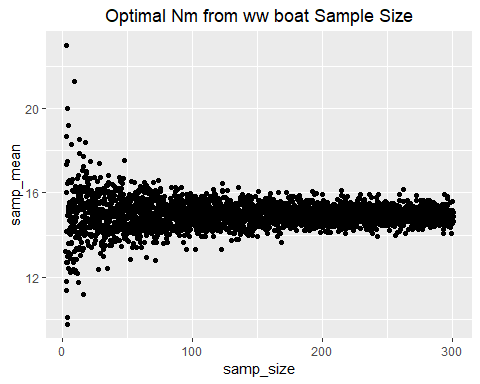
#W = 0.94807, p-value = 0.0006204  
#rejected the null hypothesis, so the data is not normal (which makes sense)

##### Figure 11: Testing assumptions of Welch two-sample t-test, Time data

### Maximum Sample Size needed to obtain simulated Means

#### To determine optimal sample size for each predicted, simulated sample sizes were run against the mean for each predictor, as show in Figures 12-15. Optimal sample sizes ranged from 100-150 for each predictor.

###Nm\_from\_boat data  
  
set.seed(2018)  
  
samp\_sim\_nm <- data.frame(  
 samp\_size = rep(3:300, time = 10)) #I want a sample from 3 through 300, ten times  
  
#take sample data frame  
samp\_sim\_nm <- samp\_sim\_nm %>%  
 #for each row (for each simulation)  
 rowwise() %>% #group by row and do something every row over and over again  
 #draw a random normal sample with n = samp\_size and take a mean  
 mutate(samp\_mean = mean(rnorm(samp\_size, mean = mean\_nm\_Y, sd= sd\_nm\_Y)))  
  
ggplot(data = samp\_sim\_nm,  
 mapping = aes(x = samp\_size, y = samp\_mean)) +  
 geom\_jitter() +  
 ggtitle("Optimal Nm from ww boat Sample Size") +  
 theme(plot.title = element\_text(hjust = 0.5))



#take simulations  
samp\_sim\_nm\_summary <- samp\_sim\_nm %>%  
 #for each sample size  
 group\_by(samp\_size) %>%  
 #calculate the SD of sample means  
 summarize(sd\_mean = sd(samp\_mean)) %>%  
 #order by SD  
 arrange(sd\_mean)

## Warning: Grouping rowwise data frame strips rowwise nature

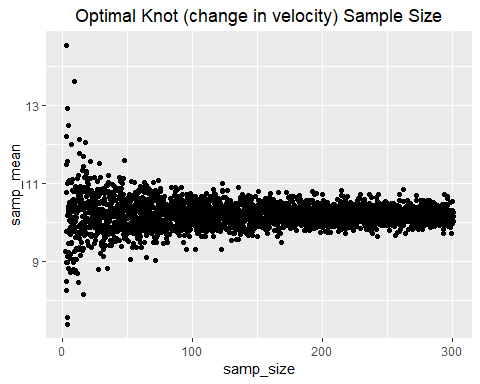
samp\_sim\_nm\_summary

## # A tibble: 298 x 2  
## samp\_size sd\_mean  
## <int> <dbl>  
## 1 297 0.143  
## 2 280 0.164  
## 3 191 0.165  
## 4 205 0.170  
## 5 149 0.180  
## 6 289 0.199  
## 7 261 0.199  
## 8 153 0.208  
## 9 264 0.212  
## 10 101 0.225  
## # ... with 288 more rows

#optimal sample size is about 100 for determining significant nm from the ww boat

##### Figure 12: Optimal Sample size for Nm\_from\_boat data. Optimal sample size is about 100 for a sample mean of 15.

###Knot data  
  
set.seed(2018)  
  
samp\_sim\_knots <- data.frame(  
 samp\_size = rep(3:300, time = 10)) #I want a sample from 3 through 300, ten times  
  
#take sample data frame  
samp\_sim\_knots <- samp\_sim\_knots %>%  
 #for each row (for each simulation)  
 rowwise() %>% #group by row and do something every row over and over again  
 #draw a random normal sample with n = samp\_size and take a mean  
 mutate(samp\_mean = mean(rnorm(samp\_size, mean = mean\_knots\_Y, sd= sd\_knots\_Y)))  
  
ggplot(data = samp\_sim\_knots,  
 mapping = aes(x = samp\_size, y = samp\_mean)) +  
 geom\_jitter() +  
 ggtitle("Optimal Knot (change in velocity) Sample Size") +  
 theme(plot.title = element\_text(hjust = 0.5))



#take simulations  
samp\_sim\_knots\_summary <- samp\_sim\_knots %>%  
 #for each sample size  
 group\_by(samp\_size) %>%  
 #calculate the SD of sample means  
 summarize(sd\_mean = sd(samp\_mean)) %>%  
 #order by SD  
 arrange(sd\_mean)

## Warning: Grouping rowwise data frame strips rowwise nature

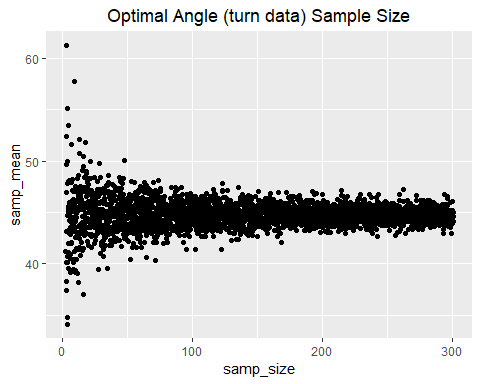
samp\_sim\_knots\_summary

## # A tibble: 298 x 2  
## samp\_size sd\_mean  
## <int> <dbl>  
## 1 297 0.0775  
## 2 280 0.0889  
## 3 191 0.0895  
## 4 205 0.0922  
## 5 149 0.0974  
## 6 289 0.108   
## 7 261 0.108   
## 8 153 0.113   
## 9 264 0.115   
## 10 101 0.122   
## # ... with 288 more rows

#optimal sample size is about 150

##### Figure 13: Optimal Sample size for change in velocity predictor data. Optimal sample size is about 150 for a sample mean of 10.

###Angle data  
  
set.seed(2018)  
  
samp\_sim\_angle <- data.frame(  
 samp\_size = rep(3:300, time = 10)) #I want a sample from 3 through 300, ten times  
  
#take sample data frame  
samp\_sim\_angle <- samp\_sim\_angle %>%  
 #for each row (for each simulation)  
 rowwise() %>% #group by row and do something every row over and over again  
 #draw a random normal sample with n = samp\_size and take a mean  
 mutate(samp\_mean = mean(rnorm(samp\_size, mean = mean\_angle\_Y, sd= sd\_angle\_Y)))  
  
ggplot(data = samp\_sim\_angle,  
 mapping = aes(x = samp\_size, y = samp\_mean)) +  
 geom\_jitter() +  
 ggtitle("Optimal Angle (turn data) Sample Size") +  
 theme(plot.title = element\_text(hjust = 0.5))



#take simulations  
samp\_sim\_angle\_summary <- samp\_sim\_angle %>%  
 #for each sample size  
 group\_by(samp\_size) %>%  
 #calculate the SD of sample means  
 summarize(sd\_mean = sd(samp\_mean)) %>%  
 #order by SD  
 arrange(sd\_mean)

## Warning: Grouping rowwise data frame strips rowwise nature

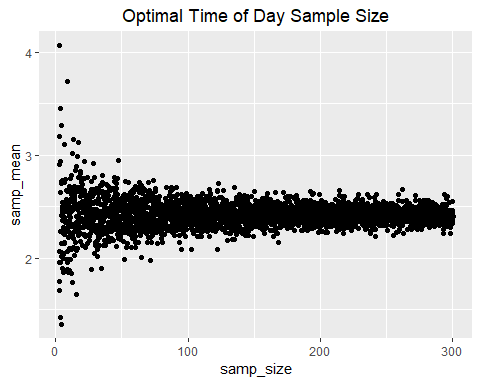
samp\_sim\_angle\_summary

## # A tibble: 298 x 2  
## samp\_size sd\_mean  
## <int> <dbl>  
## 1 297 0.294  
## 2 280 0.336  
## 3 191 0.339  
## 4 205 0.349  
## 5 149 0.369  
## 6 289 0.409  
## 7 261 0.409  
## 8 153 0.427  
## 9 264 0.434  
## 10 101 0.461  
## # ... with 288 more rows

#optimal sample size is about 100

##### Figure 14: Optimal Sample size for Angle predictor data. Optimal sample size is about 100 for a sample mean of 45.

###Angle data  
  
set.seed(2018)  
  
samp\_sim\_time <- data.frame(  
 samp\_size = rep(3:300, time = 10)) #I want a sample from 3 through 300, ten times  
  
  
#take sample data frame  
samp\_sim\_time <- samp\_sim\_time %>%  
 #for each row (for each simulation)  
 rowwise() %>% #group by row and do something every row over and over again  
 #draw a random normal sample with n = samp\_size and take a mean  
 mutate(samp\_mean = mean(rnorm(samp\_size, mean = mean\_time\_Y, sd= sd\_time\_Y)))  
  
ggplot(data = samp\_sim\_time,  
 mapping = aes(x = samp\_size, y = samp\_mean)) +  
 geom\_jitter() +  
 ggtitle("Optimal Time of Day Sample Size") +  
 theme(plot.title = element\_text(hjust = 0.5))



#take simulations  
samp\_sim\_time\_summary <- samp\_sim\_time %>%  
 #for each sample size  
 group\_by(samp\_size) %>%  
 #calculate the SD of sample means  
 summarize(sd\_mean = sd(samp\_mean)) %>%  
 #order by SD  
 arrange(sd\_mean)

## Warning: Grouping rowwise data frame strips rowwise nature

samp\_sim\_time\_summary

## # A tibble: 298 x 2  
## samp\_size sd\_mean  
## <int> <dbl>  
## 1 297 0.0293  
## 2 280 0.0336  
## 3 191 0.0339  
## 4 205 0.0349  
## 5 149 0.0369  
## 6 289 0.0408  
## 7 261 0.0409  
## 8 153 0.0426  
## 9 264 0.0434  
## 10 101 0.0460  
## # ... with 288 more rows

#optimal sample size is about 150

##### Figure 15: Optimal Sample size for Time predictor data. Optimal size is about 150 for a sample mean of 2.5.

### Power Analyses

#### Power analyses of the Welch two-sample t-tests were conducted in order to determine the minimal sample size needed to have low probability of a Type II error, or high power. The power threshold for this study was 0.80, as it is the conventional power threshold (Whitlock & Schluter 2015). Power analyses were performed on each predictor. Each analysis looked at how sample size effected power, by a factor of different standard deviations. Figure 16 results show that Nm\_from\_boat predictor would need a minimum sample size of 7 to obtain a power threshold of 0.80 (allowing a SD range of up to 8). Figure 17 results show that Knot predictor would need a minimum sample size of 43 to obtain a power threshold of 0.80 (allowing a SD range of up to 8). This higher needed sample size makes sense, as there was more simulated variation in this data. Figure 18 results show that Angle predictor would need a minimum sample size of 3 to obtain a power threshold of 0.80 (allowing a SD range of up to 8). Little variation was simulated in this predictor; hence a small sample size was expected. Lastly, in Figure 19, Time predictor will not obtain enough power, as there was no simulated difference between the control and experimental means.

set.seed(2018)  
###Nm\_from\_boat power  
  
#simulate t data  
#make a function data frame, that can repeat n amount of times  
#use a normal distribution rnorm  
  
make\_t\_data <- function(m1, m2, s, n){  
 #make a data frame, repeating treatments n number of times  
 #and use rnorm to get values  
 data.frame(treatment = c(rep("A", n), rep("B", n)),  
 value = rnorm(n\*2, mean = c(rep(m1,n), rep(m2, n)), s))  
}  
  
#test if the fuction works  
make\_t\_data(m1 = mean\_nm\_Y, m2 = mean\_nm\_N, s = sd\_nm\_Y, n = 100)

## treatment value  
## 1 A 12.73616781  
## 2 A 6.82538301  
## 3 A 14.61685856  
## 4 A 16.37562993  
## 5 A 24.05671223  
## 6 A 13.56634008  
## 7 A 25.96694590  
## 8 A 19.48325259  
## 9 A 11.75215180  
## 10 A 18.29628695  
## 11 A 11.58195792  
## 12 A 9.55209698  
## 13 A 18.69190959  
## 14 A 12.61663962  
## 15 A 16.26075112  
## 16 A 9.32044160  
## 17 A 5.37045093  
## 18 A 15.03606132  
## 19 A 6.11999189  
## 20 A 16.02727627  
## 21 A 21.58402023  
## 22 A 16.26714115  
## 23 A 16.30908966  
## 24 A 24.32028608  
## 25 A 8.55720312  
## 26 A 13.69531410  
## 27 A 9.40191089  
## 28 A 17.15468748  
## 29 A 13.53358153  
## 30 A 11.64075594  
## 31 A 13.75673707  
## 32 A 21.13721352  
## 33 A 13.55814997  
## 34 A 17.72500721  
## 35 A 5.67783288  
## 36 A 12.38575254  
## 37 A 22.35327014  
## 38 A 9.31483128  
## 39 A 16.48847178  
## 40 A 13.87080676  
## 41 A 13.06945651  
## 42 A 8.13847594  
## 43 A 10.39216112  
## 44 A 20.48514393  
## 45 A 14.17515886  
## 46 A 20.29481796  
## 47 A 11.16993797  
## 48 A 10.86995649  
## 49 A 11.71239692  
## 50 A 23.37442113  
## 51 A 11.87354891  
## 52 A 12.18825592  
## 53 A 18.08105309  
## 54 A 8.28651599  
## 55 A 13.89918296  
## 56 A 17.34469397  
## 57 A 10.05144946  
## 58 A 29.26537702  
## 59 A 16.77302469  
## 60 A 11.60988852  
## 61 A 19.33550452  
## 62 A 18.04434408  
## 63 A 15.84697897  
## 64 A 12.66175098  
## 65 A 14.53409591  
## 66 A 10.08965635  
## 67 A 22.92156441  
## 68 A 15.40447651  
## 69 A 26.24664119  
## 70 A 7.79021998  
## 71 A 16.76915036  
## 72 A 17.91928872  
## 73 A 14.92281882  
## 74 A 12.65330896  
## 75 A 18.56817444  
## 76 A 12.33524987  
## 77 A 5.15354665  
## 78 A 18.85421768  
## 79 A 14.48706290  
## 80 A 20.62572241  
## 81 A 7.79717926  
## 82 A 23.60419517  
## 83 A 10.86744793  
## 84 A 20.11137885  
## 85 A 17.60773883  
## 86 A 6.03945895  
## 87 A 28.57901392  
## 88 A 12.73788775  
## 89 A 11.10725510  
## 90 A 16.40219178  
## 91 A 9.80127792  
## 92 A 18.38918236  
## 93 A 22.68668454  
## 94 A 16.07193573  
## 95 A 6.82804810  
## 96 A 16.40863416  
## 97 A 20.11193659  
## 98 A 18.93794481  
## 99 A 11.50684627  
## 100 A 19.79561746  
## 101 B 1.94548744  
## 102 B 3.53178761  
## 103 B -3.78993452  
## 104 B -2.99435184  
## 105 B -1.41418230  
## 106 B 4.21887330  
## 107 B -6.40495926  
## 108 B 5.63470926  
## 109 B 3.41934389  
## 110 B 0.89054258  
## 111 B -2.95595911  
## 112 B -10.99261083  
## 113 B 2.57016367  
## 114 B 3.03675886  
## 115 B -2.30445694  
## 116 B 1.16303870  
## 117 B -9.26613628  
## 118 B -4.85950489  
## 119 B 15.50452209  
## 120 B 1.28653029  
## 121 B -0.82626902  
## 122 B 0.75854556  
## 123 B 1.01849148  
## 124 B 0.82284270  
## 125 B -6.76845965  
## 126 B -2.67901937  
## 127 B -4.18600565  
## 128 B -2.55979896  
## 129 B 14.21972929  
## 130 B -0.07894426  
## 131 B 0.89634292  
## 132 B -7.11532221  
## 133 B 1.26283216  
## 134 B 2.65504785  
## 135 B -5.43978662  
## 136 B 4.70250151  
## 137 B 2.15517901  
## 138 B -0.27965670  
## 139 B -1.80661019  
## 140 B -10.54498993  
## 141 B 7.15617251  
## 142 B -1.36218413  
## 143 B -0.08799716  
## 144 B 0.31091915  
## 145 B 1.83166737  
## 146 B 7.91495455  
## 147 B 3.92046000  
## 148 B 0.51526913  
## 149 B 8.91004144  
## 150 B 1.87768724  
## 151 B 0.86406287  
## 152 B 1.80820112  
## 153 B 11.95075131  
## 154 B 5.84985336  
## 155 B 3.04062579  
## 156 B -2.16089548  
## 157 B 6.54732668  
## 158 B 1.29919849  
## 159 B 3.51450234  
## 160 B -7.12641053  
## 161 B -4.33361758  
## 162 B 4.33692287  
## 163 B 9.31887168  
## 164 B 7.26578400  
## 165 B -8.21958864  
## 166 B -6.42230428  
## 167 B 5.32393294  
## 168 B 4.20066797  
## 169 B -5.99194322  
## 170 B 2.98968145  
## 171 B 8.13684816  
## 172 B 1.84099178  
## 173 B -2.87591986  
## 174 B 6.18515432  
## 175 B -3.60938804  
## 176 B -8.65077058  
## 177 B -4.90067660  
## 178 B 2.43856462  
## 179 B -5.03944690  
## 180 B -2.65503924  
## 181 B 6.29162127  
## 182 B -2.26586524  
## 183 B -0.47857781  
## 184 B -2.34041659  
## 185 B 12.42572277  
## 186 B 0.71606205  
## 187 B -5.17668679  
## 188 B 4.28321421  
## 189 B -8.26178664  
## 190 B 5.57264277  
## 191 B -0.20996034  
## 192 B -3.73087459  
## 193 B 1.39889768  
## 194 B 5.49365948  
## 195 B 2.05490637  
## 196 B 9.18078588  
## 197 B 0.83322455  
## 198 B 0.69635950  
## 199 B -5.84597409  
## 200 B 4.71674709

#the output of any t.test is a list with an entry called p.value  
#so need to extract just a p-value  
  
#extraction function:  
get\_p\_from\_t\_test <- function(sim\_data){  
 #run the t test on the data  
 test <- t.test(value ~ treatment, data = sim\_data)  
 test$p.value  
}  
  
#testing out extracton function:  
get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_nm\_Y, m2 = mean\_nm\_N, s = sd\_nm\_Y, n = 100))

## [1] 1.269181e-48

#I get a small P value with big sample size  
get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_nm\_Y, m2 = mean\_nm\_N, s = sd\_nm\_Y, n = 3))

## [1] 0.01873539

#I get a large P value the smaller the sample size I choose  
  
#now I need to simulate p values many times  
replicate(10,  
 get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_nm\_Y, m2 = mean\_nm\_N, s = sd\_nm\_Y, n = 100)))

## [1] 1.302842e-43 3.051276e-47 2.195624e-47 7.031186e-46 5.299092e-42  
## [6] 1.923511e-49 4.252505e-60 1.186866e-44 8.483529e-47 5.777869e-49

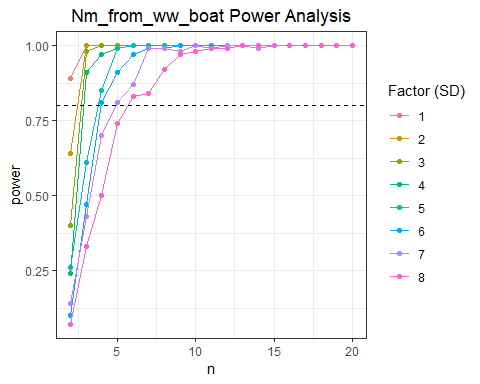
#10 P values created  
  
#Now I need some number of simulations, and a set alpha (I will do .05)  
get\_t\_power <- function(m1, m2, s, n, nsims = 100, alpha = 0.05){  
 #a vector of p values  
 p <- replicate(nsims,  
 get\_p\_from\_t\_test(make\_t\_data(m1, m2, s, n)))  
 #calculate the number of p values that are incorrect given that we should be rejecting the null  
 num\_wrong <- sum(p > alpha)  
 #return power  
 1 - num\_wrong/nsims  
}  
  
#test this function  
  
#nm\_from\_boat Power  
get\_t\_power(m1 = mean\_nm\_Y, m2 = mean\_nm\_N, s = sd\_nm\_Y, n = 100, nsims = 100, alpha=0.05)

## [1] 1

#higher power with bigger sample size  
get\_t\_power(m1 = mean\_nm\_Y, m2 = mean\_nm\_N, s = sd\_nm\_Y, n = 3, nsims = 100, alpha=0.05)

## [1] 0.57

#lower power with smaller sample size, does not reach 0.8  
  
#get power from simulations  
nm\_power <- crossing(m1 = mean\_nm\_Y, m2 = mean\_nm\_N, s = 1:8, n = 2:20) %>%  
 rowwise() %>%  
 mutate(power = get\_t\_power(m1, m2, s, n, nsim = 100)) %>%  
 ungroup()  
  
ggplot(nm\_power,  
 mapping = aes(x = n, y = power, color = factor(s))) +  
 geom\_point() +  
 geom\_line() +  
 theme\_bw() +  
 geom\_hline(yintercept = 0.8, lty =2) +  
 ggtitle("Nm\_from\_ww\_boat Power Analysis") +  
 theme(plot.title = element\_text(hjust = 0.5)) +   
 labs(colour = "Factor (SD)")



#this shows that sample sizes of at least 7 will have enough power for nm\_from\_boat data, (with a SD ranging from 1:8)

##### Figure 16: Power analysis of Nm\_from\_boat data.This shows that sample size of at least 7 will have enough power to reach the threshold of 0.8, with SD’s ranging from 1:8.

###Knot Power  
  
#testing out extracton function:  
get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_knots\_Y, m2 = mean\_knots\_N, s = sd\_knots\_Y, n = 100))

## [1] 6.753678e-26

#I get a small P value with big sample size  
get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_knots\_Y, m2 = mean\_knots\_N, s = sd\_knots\_Y, n = 3))

## [1] 0.07086268

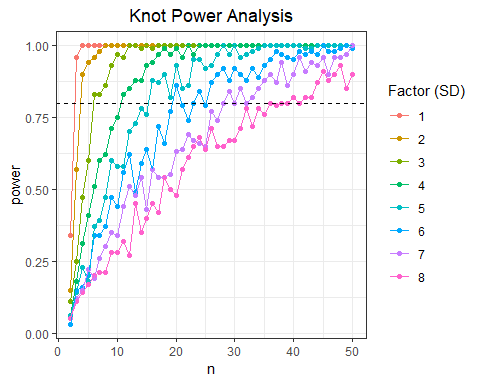
#I get a large P value the smaller the sample size I choose  
  
#test the power function  
  
#knot Power  
get\_t\_power(m1 = mean\_knots\_Y, m2 = mean\_knots\_N, s = sd\_knots\_Y, n = 100, nsims = 100, alpha=0.05)

## [1] 1

#higher power with bigger sample size  
get\_t\_power(m1 = mean\_knots\_Y, m2 = mean\_knots\_N, s = sd\_knots\_Y, n = 3, nsims = 100, alpha=0.05)

## [1] 0.27

#lower power with smaller sample size, does not reach 0.8  
  
#get power from simulations  
knot\_power <- crossing(m1 = mean\_knots\_Y, m2 = mean\_knots\_N, s = 1:8, n = 2:50) %>%  
 rowwise() %>%  
 mutate(power = get\_t\_power(m1, m2, s, n, nsim = 100)) %>%  
 ungroup()  
  
ggplot(knot\_power,  
 mapping = aes(x = n, y = power, color = factor(s))) +  
 geom\_point() +  
 geom\_line() +  
 theme\_bw() +  
 geom\_hline(yintercept = 0.8, lty =2) +  
 ggtitle("Knot Power Analysis") +  
 theme(plot.title = element\_text(hjust = 0.5)) +   
 labs(colour = "Factor (SD)")



#this shows that sample size of at least 43 to have enough power for knot\_data(with a SD ranging from 1:8)

##### Figure 17: Power analysis of Knot data. This shows that sample size of at least 43 will have enough power to reach the threshold of 0.8, with SD’s ranging from 1:8.

###Angle Power  
  
#testing out extracton function:  
get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_angle\_Y, m2 = mean\_angle\_N, s = sd\_angle\_Y, n = 100))

## [1] 1.888688e-74

#I get a small P value with big sample size  
get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_angle\_Y, m2 = mean\_angle\_N, s = sd\_angle\_Y, n = 3))

## [1] 0.007126051

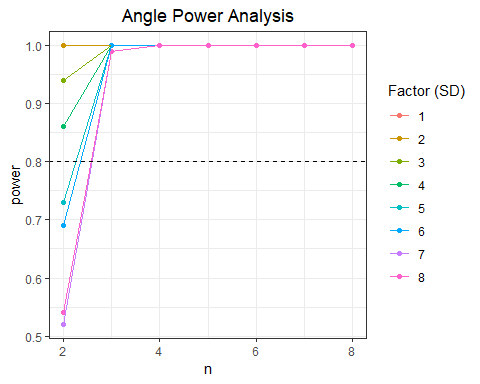
#I get a large P value the smaller the sample size I choose  
  
#test the power function  
  
#angle Power  
get\_t\_power(m1 = mean\_angle\_Y, m2 = mean\_angle\_N, s = sd\_angle\_Y, n = 100, nsims = 100, alpha=0.05)

## [1] 1

#higher power with bigger sample size  
get\_t\_power(m1 = mean\_angle\_Y, m2 = mean\_angle\_N, s = sd\_angle\_Y, n = 3, nsims = 100, alpha=0.05)

## [1] 0.87

#lower power with smaller sample size, just over 0.8  
  
#get power from simulations  
angle\_power <- crossing(m1 = mean\_angle\_Y, m2 = mean\_angle\_N, s = 1:8, n = 2:8) %>%  
 rowwise() %>%  
 mutate(power = get\_t\_power(m1, m2, s, n, nsim = 100)) %>%  
 ungroup()  
  
ggplot(angle\_power,  
 mapping = aes(x = n, y = power, color = factor(s))) +  
 geom\_point() +  
 geom\_line() +  
 theme\_bw() +  
 geom\_hline(yintercept = 0.8, lty =2) +  
 ggtitle("Angle Power Analysis") +  
 theme(plot.title = element\_text(hjust = 0.5)) +   
 labs(colour = "Factor (SD)")



#this shows that sample size of at least 3 would have enough power for angle data - because the mean\_angle\_N is basically 0.

##### Figure 18: Power analysis of Angle data. This shows that sample size of at least 3 will have enough power to reach the threshold of 0.8, with SD’s ranging from 1:8.

###Time Power  
  
#testing out extracton function:  
get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_time\_Y, m2 = mean\_time\_N, s = sd\_time\_Y, n = 100))

## [1] 0.4796852

#no significant p value (as expected)  
get\_p\_from\_t\_test(make\_t\_data(m1 = mean\_time\_Y, m2 = mean\_time\_N, s = sd\_time\_Y, n = 3))

## [1] 0.8488812

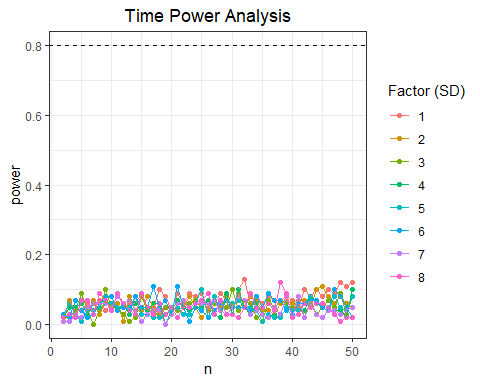
#no significant p value (as expected)  
  
#test the power function  
  
get\_t\_power(m1 = mean\_time\_Y, m2 = mean\_time\_N, s = sd\_time\_Y, n = 100, nsims = 100, alpha=0.05)

## [1] 0.11

#higher power with bigger sample size  
get\_t\_power(m1 = mean\_time\_Y, m2 = mean\_time\_N, s = sd\_time\_Y, n = 3, nsims = 100, alpha=0.05)

## [1] 0.02

#lower power with smaller sample size, just over 0.8  
  
#get power from simulations  
time\_power <- crossing(m1 = mean\_time\_Y, m2 = mean\_time\_N, s = 1:8, n = 2:50) %>%  
 rowwise() %>%  
 mutate(power = get\_t\_power(m1, m2, s, n, nsim = 100)) %>%  
 ungroup()  
  
ggplot(time\_power,  
 mapping = aes(x = n, y = power, color = factor(s))) +  
 geom\_point() +  
 geom\_line() +  
 theme\_bw() +  
 geom\_hline(yintercept = 0.8, lty =2) +  
 ggtitle("Time Power Analysis") +  
 theme(plot.title = element\_text(hjust = 0.5)) +   
 labs(colour = "Factor (SD)")



#since there is no simulated difference between the time means, not being able to reach a power threshold is expected

##### Figure 19: Power analysis of Time data. As there is no significant difference between the two means, the power threshold of 0.8 will not be able to be met.

# DISCUSSION

#### According to the power analyses, significant results of the Welch two-sample t-tests can be achieved with relatively small sample sizes, assuming that real-time data follows trends that were simulated. Real-time AIS data will be processed for this study for future analysis. When real-time AIS data are obtained, I will most likely have uneven sample sizes, so statistical testing will need to account for that (a Welch two-sample t-test accounts for unequal variances, so would still be appropriate). Normality of the real-time data will also need to be investigated and tested, in order to meet assumptions for the Welch two-sample t-test. If the normal distribution assumption of the real-time data cannot be met, a Mann-Whitney U-test would be used in its place, as it can be used to compare two means with non-normal distribution (Whitlock & Schluter 2015).

#### When analyzing real-time AIS data, the predictors will need to be tested to see if they are actually independent of each other. Plots comparing predictors can be used as they were in the initial visualizations shown in Figure 3. If all or some predictors appear dependent of each other, ANOVA statistical tests could be used to investigate potential covariance.

#### Finally, to do determine a true “control” group, I will need to see if there are other reasons that ships might deviate from their track other than if whale watch boats via AIS. When whale watch boats are not shown to be present on AIS, I will sub-sample some control days to see how often tracks deviate from their track and/or change velocity. To further confirm a true “control”, I will use my multi-year whale watch database with Boston Harbor Cruises to additionally confirm whether whale watch boats were present (and not reporting location via AIS). Using this data, I can see if ships also deviate tracks when non-AIS reporting whale watch boats are present.

# REFERENCES

#### Davis, G.E., Baumgartner, M.F., Bonnell, J.M., Bell, J., Berchok, C., Thornton, J.B., Brault, S.,Buchanan, G., Charif, R.A., Cholewiak, D. and Clark, C.W., 2017. Long-term passive acoustic recordings track the changing distribution of North Atlantic right whales (Eubalaena glacialis) from 2004 to 2014. Scientific Reports, 7(1), p.13460.

#### Federal Register (2003) Automatic Identification System; vessel carriage requirement. Coast Guard, U.S. Department of Homeland Security, 60559-60570. October 22.

#### International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA) (2004) IALA Guideline No. 1028 on the Automatic Identification System (AIS): Part I. Operational issues. 1. Saint Germain en Laye, France.

#### Nowacek, D.P., M.P. Johnson, and P.L. Tyack. 2004. North Atlantic right whales (Eubalaena glacialis) ignore ships but respond to alerting stimuli. Proceedings of the Royal Society of London B: Biological Sciences 271:227-231.

#### Thomas, P.O., R.R. Reeves, and R.L. Brownell. 2016. Status of the world’s baleen whales. Marine Mammal Science 32:682-734.

#### Van Der Hoop, J.M., M.J. Moore, S.G. Barco, T.V. Cole, P.Y. Daoust, A.G. Henry, D.F. McAlpine, W.A. McLellan, T. Wimmer, and A.R. Solow. 2013. Assessment of management to mitigate anthropogenic effects on large whales. Conservation Biology 27:121-133.

#### Whitlock, M & D. Schluter. 2015. Roberts and Company Publishers. Greenwood Village, CO.

#### Wiley, D.N, L. Hatch, M. Thompson, K. Schwehr, and C. MacDonald. 2013. Marine Sanctuaries and Marine Planning: Protecting endangered marine life. National Ocean Policy. Proceedings Fall 2013:10-15. www.uscg.mil/proceedings